

# Todo

Año 2 • Número 14 • 6,90 euros



# Programación

La Revista bimestral para entusiastas de la programación

www.studiopress.es



**NUEVO**

## Visual Studio .NET 2005

Depurar errores de gestión de memoria en Linux con **Valgrind**

**OpenGL:** diseñando nuestro primer programa paso a paso

Programación práctica en **Qt:** un mezclador de sonidos

Acceso a bases de datos en PHP mediante **ADOdb**

■ CD-ROM: ECLIPSE 3.1.1 + PLUGINS PARA LINUX Y WINDOWS, STRUTS 1.2.7, NEMERLE 0.9.0, VALGRIND 3.0.1...

### Zona Linux

- **Nemerle:** Nuevo curso "desde cero"
- **Mono:** Así se programará con C# 3.0



### Zona Java

- Introducción al framework **Struts**
- Creación de ayudas Java online con **JavaHelp**



**DOSSIER: ECLIPSE 3.1 Y SUS HERRAMIENTAS PARA C/C++**



## DIRECTOR

Eduardo Torbio  
etorbio@iberprensa.com

## REDACCIÓN

Yenifer Trabadela  
yenifer@iberprensa.com

## COLABORADORES

Antonio M. Zugaldía  
(azugaldia@iberprensa.com)  
David Santo Orcero  
(orcero@iberprensa.com)  
Manuel Domínguez  
(mdominguez@iberprensa.com)  
Fernando Escudero  
(fescudero@iberprensa.com)  
José Manuel Navarro  
(jnavarro@iberprensa.com)  
Marcos Prieto  
(mprieto@iberprensa.com)  
Guillermo "el Guille" Som  
(elguille@iberprensa.com)  
Santiago Márquez  
(samarquez@iberprensa.com)  
José Rivera  
(jrivera@iberprensa.com)  
Alejandro Serrano  
(aserrano@iberprensa.com)  
Jordi Massaguer  
(jmassaguer@iberprensa.com)  
Pedro Agulló  
(pagullo@iberprensa.com)  
Moisés Díaz  
(mdiaz@iberprensa.com)  
Fernando Marín  
(fmarin@iberprensa.com)  
Fabian Seoane  
(fseoane@iberprensa.com)

## DISEÑO PORTADA Y MAQUETACIÓN

Antonio G. Tomé

## DIRECTOR DE PRODUCCIÓN

Carlos Peropadre  
cperopadre@iberprensa.com

## ADMINISTRACIÓN

Marisa Cogono

## SUSCRIPCIONES

Tel: 91 628 02 03  
suscripciones@iberprensa.com

## FILMACIÓN: Fotpreim Duvial

IMPRESIÓN: I. G. Printone  
DUPLICACIÓN CD-ROM: M.P.O.

## DISTRIBUCIÓN

S.G.E.L.  
Avda. Valdelaparra 29 (Pol. Ind.)  
28108 Alcobendas (Madrid)  
Tel.: 91 657 69 00

## EDITA: Studio Press

www.studiopress.es



## REDACCIÓN, PUBLICIDAD Y ADMINISTRACIÓN

C/ del Río Ter, Nave 13.  
Polígono "El Nogal"  
28110 Algete (Madrid)  
Tel.: 91 628 02 03\*  
Fax: 91 628 09 35  
(Añada 34 si llama desde fuera de España.)

*Todo Programación* no tiene por qué estar de acuerdo con las opiniones escritas por sus colaboradores en los artículos firmados. Los contenidos de *Todo Programación* son copropiedad de Iberprensa y sus respectivos autores.

Iberprensa es una marca registrada de Studio Press.

## DEPÓSITO LEGAL: M-13679-2004

Número 14 • Año 2  
Copyright 1/01/06  
PRINTED IN SPAIN

# EDITORIAL



Eduardo Torbio

## Actualidad manda

Muchas novedades se han producido estos últimos días en nuestro pequeño gran mundo del desarrollo, y tratamos en este número de dar cobertura a todas ellas. Por ejemplo, Visual Studio 2005, donde de la mano de Guillermo Som nos adentramos en las nuevas características del entorno de Microsoft. También en la sección Mono dedicamos un completo reportaje a analizar técnicamente algunas de las nuevas características del lenguaje C# en su versión 3.0. Pero no acaba aquí la parte que dedicamos en este número de **Todo Programación** a la actualidad, ya que además tenemos nueva versión de Eclipse, la 3.1, con una sustancial mejora, sobre todo a nivel de funcionalidad, y algunas novedades en lo que se refiere a capacidades. Veremos también cómo marcha el plugin para trabajar en C/C++. En cuanto a la parte más práctica de la revista, hemos elegido una serie de tutoriales orientados a programadores en Java y a herramientas más específicas de la plataforma Linux. Tanto Java como Linux tendrán ya desde este número y en lo sucesivo una mayor presencia en la revista. Digamos que, junto a .NET, configuran el esqueleto sobre el cual elaboramos nuestros contenidos. Esperemos que os gusten, de momento, ya somos la revista especializada en desarrollo de mayor difusión en España, es un hecho y no una frase hecha.

## SUSCRIPCIONES

Como oferta de lanzamiento existe la posibilidad de suscribirse durante 12 números a **Todo Programación** por solo 66,30 euros lo que significa un ahorro del 15% respecto el precio de portada. Además de regalo puedes elegir entre una de las dos guías que aparecen abajo. Más información en: [www.studiopress.es](http://www.studiopress.es)



## SERVICIO TÉCNICO

**Todo Programación** dispone de una dirección de correo electrónico y un número de Fax para formular preguntas relativas al funcionamiento del CD-ROM de la revista.  
e-mail: [todoprogramacion@iberprensa.com](mailto:todoprogramacion@iberprensa.com)  
Fax: 91 628 09 35

## LECTORES

Comparte con nosotros tu opinión sobre la revista, envíanos tus comentarios, sugerencias, ideas o críticas.

**Studio Press**  
(**Todo Programación**)  
C/ Del Río Ter, Nave 13  
Pol. "El Nogal"  
28110 Algete. Madrid

## DEPARTAMENTO DE PUBLICIDAD

Si le interesa conocer nuestras tarifas de publicidad no dude en ponerse en contacto con nuestro departamento comercial:

Tel. 91 628 02 03  
e-mail: [publicidad@iberprensa.com](mailto:publicidad@iberprensa.com)





Número 14

### A quién vamos dirigidos

**Todo Programación (TP)** es una revista para programadores, escrita por programadores y con un enfoque eminentemente práctico. Trataremos de ser útiles al programador, tanto al profesional como al estudiante. Si hay algo cierto en este sector es que nunca podemos parar, vivimos en un continuo proceso de reciclaje. Ahí es donde tratará de encajarse TP: información, actualidad, cursos y prácticas de los lenguajes más demandados y formación en sistemas.



## En portada

# Dossier: El nuevo Visual Studio 2005

**10** Desde la aparición de la primera versión de Visual Studio .NET todo lo relacionado con la programación ha cambiado debido al gran impacto que esta plataforma ha tenido en el mercado. Ahora con la nueva versión que se presenta, la 2005, se refuerza y se facilita la creación de aplicaciones basadas en la nueva versión de .NET Framework 2.0. En el dossier que hemos elaborado analizamos las nuevas características que convierten a este entorno en una herramienta muy útil a la hora de desarrollar aplicaciones de escritorio, web o dispositivos móviles.



## ZONA LINUX

### Actualidad: Eclipse 3.1 y sus herramientas para C/C++ >>

**16** Eclipse 3.1 supone la culminación de un año de trabajo de la Fundación Eclipse, una de las uniones más prósperas entre software libre, empresa y Java. Para esta última versión no se han introducido cambios radicales, pero sí una serie de pequeñas mejoras que hacen el trabajo con esta herramienta mucho más productivo.

### Avance de las nuevas características para C# 3.0 >>

**19** Recientemente Microsoft ha publicado un documento donde se describen las principales novedades que incorporará C# en su versión 3.0. Dedicamos este mes nuestra habitual sección de desarrollo Mono a analizar estas nuevas características.



## .net WINDOWS

### Trucos C++: Lanzando aplicaciones >>

**23** Dedicamos un tutorial para ver de manera muy rápida y práctica cómo lanzar y ejecutar aplicaciones desde nuestros programas en C++ y es que son muchas las ocasiones en que necesitamos realizar ciertas operaciones sin delegar en el sistema operativo.



### Ejemplos y código fuente

Cada CD-ROM de la revista incluye una carpeta denominada **fuentes** en la que se encuentra el material complementario para seguir cada uno de los cursos: por ejemplo, los listados completos, los ejemplos desarrollados en diversos lenguajes, compiladores, editores, utilidades y, en general, cualquier herramienta que se mencione o cite en la respectiva sección. Todo con la finalidad de completar la formación y facilitar el seguimiento de cada artículo por parte del lector.

## CONTENIDO DEL CD-ROM

### Herramientas y recursos para el programador

**64** En el CD de **Todo Programación** puedes encontrar en cada número las herramientas de programación más útiles y que conseguirán simplificar tu trabajo. Además, tendrás a tu disposición todas las aplicaciones y los ejemplos que se requieran para seguir correctamente los artículos y cursos que presentamos en la revista. En este número el protagonista es Eclipse y sus plugins.





# TALLER PRÁCTICO



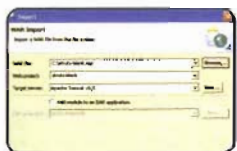
## Acceso a bases de datos en PHP mediante ADOdb >>

**36** ADOdb es un conjunto de clases PHP con soporte para multitud de sistemas gestores de bases de datos, que permite abstraer al programador del acceso a los datos de una manera similar al que se produce con JDBC en Java.



## Programación multimedia en Qt : mezclador de sonidos >>

**40** Vamos a realizar una práctica para seguir progresando en nuestros conocimientos de las librerías multiplataforma Qt, en esta ocasión vamos a estudiar la programación de un mezclador de sonido, nos apoyaremos también en el uso de OpenGL.



## Introducción al framework Struts >>

**44** En dos entregas vamos a abordar el que probablemente se ha convertido en el estándar de desarrollo de aplicaciones web sobre Java, Struts. Introducimos esta herramienta a nivel teórico y veremos un primer contacto práctico.



## Creación de ayudas Java online con JavaHelp >>

**49** Es de vital importancia que seamos capaces de construir un buen sistema de ayuda para el usuario que utilizará la aplicación que desarrollamos. Si lo hacemos en Java, Sun pone a nuestra disposición JavaHelp, descubramoslo



## OpenGL: nuestro primer programa >>

**53** En nuestro anterior número tuvimos la primera toma de contacto con OpenGL. Ahora vamos a realizar nuestro primer programa para ir viendo los mecanismos fundamentales para trabajar con esta librería multiplataforma.



## Valgrind, localizando errores de gestión de memoria >>

**57** Existen muchas aplicaciones que analizan código de forma estática y depuradores que funcionan a buen nivel, pero no detectan errores causados por el uso incorrecto de la memoria dinámica. Es ahí donde entra en juego el sistema de depuración Valgrind.

## NOTICIAS

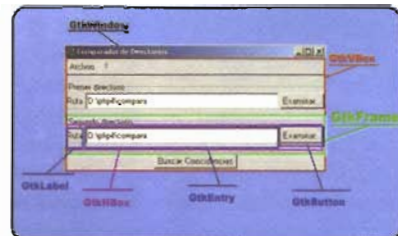
6. Madrid acoge la sesión inaugural del HP Integrity Summit 2005.
6. Nueva edición de BEA World en California.
7. IBM y Red Hat se alían para impulsar Linux en los nuevos mercados.
7. CA cede patentes a la comunidad open source.
8. Oracle Open World.
8. SGI incorpora tecnología reconfigurable RASC.
8. HP firma acuerdos con la ONCE y la SGFD.
8. Aula SUN en la UC3M.
9. Archos lanza sus nuevos reproductores multimedia portátiles.
9. Nuevo portátil Airis PRAGMA 845.
9. Equipos multifunción Dell All-in one.



# Y ADEMÁS...

## DESARROLLO WEB: Diseño con PHP-GTK >>

**62** PHP-GTK es una extensión para PHP que nos permite la posibilidad de crear interfaces gráficas de usuario (GUI) para que nuestras aplicaciones interactúen con el usuario a través de su entorno gráfico (Windows, KDE, GNOME, etc.)



## CUADERNOS DE PRINCIPIANTES

### Introducción a Nemerle >>

**28** Nemerle es un lenguaje de alto nivel para la plataforma .NET



que vamos a utilizar en este curso para principiantes. El objetivo es aprender a programar para .NET de una manera sencilla, a nivel práctico, paso a paso aprovechando las virtudes de este lenguaje.

### Visual Basic para Aplicaciones: macros en Excel >>

**33** Excel es una aplicación muy extendida y usada, no hay duda, pero vamos a dar un salto importante en su manejo mediante la introducción de macros que nosotros mismos podemos programar utilizando Visual Basic para Aplicaciones.







# Creación de ayudas Java online con JavaHelp

MANUEL DOMÍNGUEZ

mdominguez@iberpresa.com



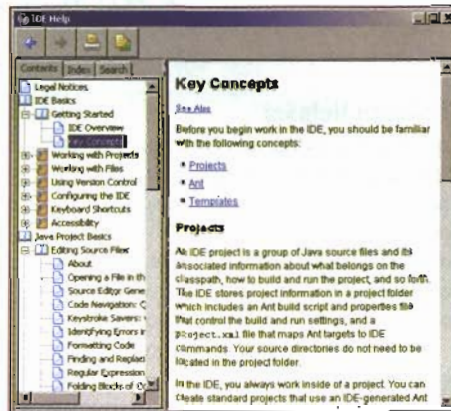
En la mayoría de las ocasiones, ocupamos gran parte del tiempo en hacer un software adecuado y dejamos de lado la construcción de un buen sistema de ayuda para el usuario que finalmente utilizará dicho software.

Si desarrollamos en Java, Sun pone a nuestra disposición una extensión de J2SE llamada JavaHelp, gracias a la cual podremos crear vistosos sistemas de ayuda e integrarlos en nuestra aplicación.

## ¿QUÉ ES JAVAHELP?

JavaHelp es un conjunto de clases desarrollada por Sun, una extensión de J2SE cuyo cometido es facilitar al desarrollador de aplicaciones Java la creación de ayudas online para sus programas. Suele dedicarse poco tiempo a los sistemas de ayuda en el software, al menos en aquellos sistemas de tamaño reducido; muchas veces se incluyen tediosos documentos PDF junto a la aplicación para que el usuario los consulte, pero no es lo recomendable. Lo realmente atractivo es que éste pueda consultar desde la propia aplicación, y de una forma más o menos interactiva, aquello que necesite. Además, hasta hace relativamente poco, los sistemas de ayuda diferían unos de otros, por lo que tampoco se seguía un criterio común y esto daba muy mala imagen.

Con JavaHelp, además de desarrollar un sistema de ayuda fácilmente, podemos mostrarla en nuestra aplicación directamente y de una forma uniforme en todos los desarrollos Java. Y, gracias al navegador de ayuda que proporciona la API de JavaHelp, podremos acceder directamente al apartado que interesa, realizar búsquedas en la ayuda, por contenido, por glosario, etcétera. Mucho más cómodo para nuestros clientes



Ejemplo del navegador de JavaHelp.

y a la larga beneficioso para nuestra imagen (y para nuestra cuenta corriente).

## REQUISITOS

JavaHelp utiliza diversas tecnologías bien combinadas para proporcionar todo lo necesario: HTML, Java, XML..., así que es preciso tener al menos conocimientos sobre estos temas. No es imprescindible al cien por cien, porque hay herramientas como JavaHelpDev (mostrada al final del artículo e incluida en el CD de la revista) que facilitan la labor, pero cuanto menos es bastante recomendable. Además, debemos contar con el paquete que contiene las clases de JavaHelp, que se puede descargar de <http://java.sun.com> (se incluye en el CD de la revista) y que actualmente esté en su versión 2.0.

Durante todo el artículo vamos a utilizar NetBeans 4.1 para desarrollar los ejemplos, aunque se puede utilizar cualquier otro IDE. Por último, JavaHelp necesita que el JSDK instalado sea al menos la versión 1.2.2; no debería ser un problema porque prácticamente nadie utiliza hoy en día versiones inferiores a ésta, pero no está de más advertirlo.

Por tanto, supondremos que el lector está familiarizado con el desarrollo en Java, usa un IDE cualquiera que ya tiene configurado y tiene instalada una versión de J2SE adecuada.

## CONFIGURACIÓN DEL ENTORNO

JavaHelp está compuesto por diversos ficheros. Cuatro de los más importantes se encuentran en `JAVAHELP_HOME/javahelp/lib` y se llaman: `jh.jar`, `jhbasic.jar`, `jhall.jar` y `jsearch.jar`. Todos son conjuntos de librerías que proporcionan al desarrollador diversos servicios. El primero de los ficheros es la API estándar de JavaHelp, incluye el navegador JavaHelp básico con las funciones de búsqueda, índice y tabla de contenidos. El segundo es un subconjunto del primero, pero no incorpora clases para proporcionar búsqueda completa en texto. El tercero contiene todas las clases existentes de JavaHelp, además de las herramientas necesarias para el despliegue del sistema de ayuda. El último archivo incluye solo el sistema de búsqueda en la ayuda de JavaHelp. Al menos uno de los ficheros anteriores debe estar disponible para poder hacer uso de toda o parte de la API de JavaHelp.

En definitiva, los pasos que hay que seguir son los que explicamos a continuación. En primer lugar descomprimos el paquete de JavaHelp, que es un fichero ZIP, en el lugar deseado. Posteriormente debemos crear una variable de entorno llamada `JAVAHELP_HOME` que apunte al directorio que se ha creado tras la descompresión del paquete anterior. También es necesario incluir en la variable de entorno `PATH` la ruta `JAVAHELP_HOME/javahome/bin` para poder acceder a las utilidades que generan índices de búsqueda para la ayuda y, por último, hacer que uno de los cuatro ficheros mencionados unas líneas atrás, o todos, estén en la variable de entorno `CLASSPATH`. Lo normal es incluir en el `CLASSPATH` el fichero `jh.jar`, aunque esto depende de lo que necesitemos y el deseo del desarrollador.

Con esto habremos terminado de configurar el entorno, y tanto las clases JavaHelp

JavaHelp es un conjunto de clases para facilitar al desarrollador la creación de ayudas online



como las utilidades necesarias estarán disponibles para poder comenzar a desarrollar.

## ■ ESTRUCTURA DE UN PROYECTO JAVAHHELP

Un proyecto JavaHelp está formado por diversos componentes, cada uno de los cuales tiene una misión específica y debe ser generado de una forma concreta. Crear una ayuda JavaHelp no implica que haya que programar algo en Java. Realmente solo será necesario tocar código Java para añadir dicho proyecto JavaHelp, esto es, la ayuda que creemos, en la aplicación. En las siguientes líneas se desglosan estos componentes y el modo en que interactúan unos con otros.

### La documentación de ayuda

JavaHelp presenta en el navegador de la ayuda una serie de documentos HTML que forman la ayuda online en sí. Así pues es imprescindible tener un conjunto de páginas

web con el contenido que se mostrará al usuario. JavaHelp permite posteriormente clasificar e indizar dicha ayuda para mostrarla al usuario de una forma más eficiente, y para que éste puede navegar a través de ella según las técnicas que ya se han comentado. Por rutina y por comodidad, parece lógico crear dentro de nuestro proyecto una carpeta *help* o *ayuda* y dentro de ella incluir todas las imágenes y páginas HTML que deseemos. La única restricción es que el HTML utilizado ha de ser HTML 3.2 o posterior, pero eso hoy en día no es un problema demasiado grande.

### El fichero HelpSet

Este fichero contiene la descripción en XML de todo el sistema de ayuda. Cuando se invoca a dicho sistema, al navegador de JavaHelp, se analiza este archivo antes que ninguno para poder tener localizado todo el contenido de la ayuda y mostrar en el navegador las opciones correspondientes.

Entre otras cosas, un fichero HelpSet debe especificar el archivo de mapeado del sistema de ayuda, información de cómo ha de mostrarse el navegador de la ayuda JavaHelp al usuario, y el título de la ayuda o la pantalla que aparecerá por defecto en el navegador antes de que el usuario seleccione otras opciones.

Como todo fichero XML, debe tener una etiqueta raíz, que en este caso es `<helpset>`. Dentro de ella se definen diversos aspectos, por ejemplo, el título que tendrá la ventana de ayuda, entre las etiquetas `<title>`, etcétera.

Explicar en este artículo todas las etiquetas que puede llevar un fichero HelpSet es complicado. Dentro del propio ZIP que bajamos con las clases de JavaHelp 2.0, hay un documento PDF que las trata con detalle (punto 4.4 del PDF). Comentaremos sin embargo qué se puede definir. En primer lugar, se debe definir la localización del fichero de mapeado, en forma de URL o de fichero local; además, hay que decir también cuál

será la página de ayuda que se verá en un inicio. Posteriormente tenemos que determinar un conjunto de etiquetas `<VIEW>` que parametrizan el navegador de ayuda. Es ahí donde decidimos si queremos que aparezca una pestaña con la tabla de contenidos, un cuadro para buscar, un índice o cualquier combinación de ellos. En JavaHelp 2.0 se permiten además opciones como búsqueda por glosario. Para cada uno de estos elementos XML hay que definir su nombre, etiqueta (la que se verá en pantalla); el fichero XML que lo define (se verán en los siguientes puntos) y la clase JavaHelp encargada de mostrarlo; algo como `javax.help.TOCView`, por ejemplo, si se trata de una tabla de contenidos. Además, se puede especificar el tamaño que tendrá la ventana del navegador de ayuda JavaHelp.

Será más sencillo viéndolo con un ejemplo concreto. En nuestro caso vamos a crear una ayuda cuyo fichero HelpSet asociado se puede ver en el **Listado 1**.



Ejemplo sencillo para incorporar JavaHelp a una aplicación.

#### Listado 1. Fichero HelpSet (primeraayuda.hs)

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<!DOCTYPE helpset PUBLIC
"-//Sun Microsystems Inc.
//DTD JavaHelp HelpSet Version 1.0
//EN" "http://java.sun.com/
products/javahelp/helpset_1_0.dtd">
<helpset version="1.0">
  <title>Help</title>
  <maps>
    <homeID>top</homeID>
    <mapref location="Map.jhm"/>
  </maps>
  <view>
    <name>TOC</name>
    <label>Table of Contents</label>
    <type>javax.help.TOCView</type>
    <data>primeraayudaTOC.xml</data>
  </view>
  <view>
    <name>Index</name>
    <label>Index</label>
    <type>javax.help.IndexView</type>
    <data>primeraayudaIndex.xml</data>
  </view>
  <view>
    <name>Search</name>
    <label>Search</label>
    <type>javax.help.SearchView</type>
    <data engine="com.sun.java.help.
search.DefaultSearchEngine">
      JavaHelpSearch</data>
  </view>
</helpset>
```

#### Listado 2. Fichero de mapeado (Map.jhm)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE map PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp Map Version
1.0//EN" "http://java.sun.com/products/javahelp/map_1_0.dtd">
<map version="1.0">
  <mapID target="top" url="principal.html"/>
  <mapID target="apartado3" url="apartado3.html"/>
  <mapID target="apartado1" url="apartado1.html"/>
  <mapID target="principal" url="principal.html"/>
  <mapID target="apartado4" url="apartado4.html"/>
  <mapID target="apartado2" url="apartado2.html"/>
</map>
```



do el identificador único que queremos dar a ese elemento del fichero de mapeado y URL, que es la dirección del fichero al que equivaldrá dicho *target*. En el resto de archivos solo se hará referencia al *target* descrito aquí cuando se desee utilizar el fichero al que apunta el parámetro URL. Por ejemplo:

```
<mapID target="UsarJavaHelp"
url="html/usarjavahelp.html" />
```

Asocia al fichero *html/usarjavahelp.html* el identificador *UsarJavaHelp* y a partir de ese momento cada vez que queramos hacer referencia al fichero HTML, usaremos este identificador. Para seguir con nuestro ejemplo, hemos creado un fichero de mapeado que se puede observar en el **Listado 2** de la página anterior.

Éste es uno de los procesos más tediosos de la creación de la ayuda, puesto que si tenemos un sistema complejo, necesitaremos muchas entradas en este fichero para asociar a cada archivo HTML (u otros) que deseemos referenciar, con un identificador único.

### La tabla de contenidos

El fichero donde se define la tabla de contenidos permite construir el árbol de contenidos que mostrará el navegador JavaHelp. Cuando creamos el fichero de mapeado, asociamos a una etiqueta una URL concreta. En este fichero asociamos a las mismas etiquetas un texto que se muestre en pantalla, de forma que sea fácil seguir la ayuda para el usuario. Es decir, internamente el fichero de mapeado y el de contenidos funcionan con los identificadores que nosotros definimos en el primero, como si fuesen nombres de variables o campos de una tabla, pero no podemos mostrarlos en el árbol del GUI sino que se sustituirán por el texto que definamos en este fichero.

En este caso, hay varias etiquetas posibles con sus respectivos atributos para el fichero de contenidos (de nuevo un archivo XML). Las más comunes son *<toc>*/*</toc>*,

dentro de las cuales deben estar todas las demás que deseemos poner, y *<tocitem>* que contiene dos parámetros: *text* y *target*. El primero indicará al navegador JavaHelp el texto que debe mostrar en la tabla de contenidos, mientras que *target*, uno de los identificadores que definimos en el fichero de mapeado, permitirá al navegador JavaHelp acceder mostrar un fichero de ayuda (HTML) cuando se haga clic sobre esta entrada TOC en el árbol de contenidos. Suele tener un aspecto como:

```
<tocitem text="Tema 1: crear
facturas" target="crearfactura" />
```

donde se supone que *crearfactura* está definido en el fichero de mapeado y asociado a un documento HTML. Nuestro ejemplo también contiene, por supuesto, un fichero de definición de la tabla de contenidos que se puede observar en el **Listado 3**. Realmente se trata de un ejemplo muy sencillo, pero sirve para aprender las nociones básicas. Complicar más el ejemplo solo implica añadir más etiquetas de cada tipo, pero quizás no merezca la pena enmarañarlo tanto.

### El fichero de índice

El fichero de índice indica al navegador de JavaHelp el contenido y la forma del índice de navegación de la ayuda. Consiste en un fichero creado en XML, como los anteriores, donde se indican los items de ayuda que aparecerán en cada sección del índice. Es parecido al archivo de descripción de la tabla de contenidos. Realmente hay un fichero XML por cada vista *<VIEW>* que definimos en el fichero HelpSet, porque para cada una de las pestañas u opciones que permite el navegador, si se activan, se han de definir los contenidos que mostrará.

En este caso, hay un elemento XML principal *<index>*/*</index>* dentro del cual se agrupan otros, de forma muy similar a la tabla de contenidos. Los "hijos" que hay que crear son las entradas del índice, lo que se hace con etiquetas *<indexitem>* que, de nuevo,

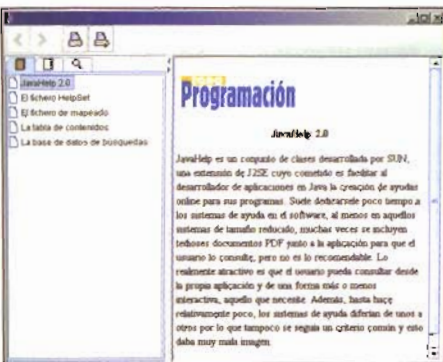
tienen los dos atributos *text* y *target*, que cumplen la misma función que en la descripción de la tabla de contenidos. Por ejemplo:

```
<indexitem text="Base de datos "
target="bd" />
```

mostraría en la pestaña de índice del navegador JavaHelp el texto *Base de datos* y al hacer clic sobre ese texto, aparecería el fichero de ayuda que se hubiese asociado en el fichero de mapeado al identificador *bd*. El fichero de índice del ejemplo que estamos siguiendo se muestra en el **Listado 4** de la página siguiente.

### Búsquedas

Para dotar de búsquedas a nuestro sistema de ayuda, no hace falta crear un fichero XML asociado, puesto que los datos que habna que insertar dentro serían muchos. En su lugar es necesario utilizar una herramienta adicional llamada *jindexer* que viene en el ZIP de JavaHelp que descargamos de Javasoft. Dado que las búsquedas de este tipo requieren un análisis exhaustivo de los documentos sobre los que trabajar, se realiza la operación en dos pasos. Primero la herramienta realiza una indización conceptual de los documentos que componen la ayuda, localizando términos existentes y asociándolos al mismo documento y a su posición dentro de él. Con esto se genera una base de datos de búsqueda precalculada, que se almacena dentro del directorio donde tengamos los HTML de ayuda, en una carpeta llamada *JavaHelpSearch*. Sobre esa base de datos, la API de JavaHelp realiza una búsqueda basándose en análisis del lenguaje natural de la consulta introducida y en un rastreo sobre la propia base de datos, ofreciendo al usuario presentaciones de la ayuda acorde a lo que haya preguntado/buscado. Es una operación bastante compleja, pero afortunadamente nosotros como creadores de la ayuda solo tenemos que crear la base de datos, lo cual es automático gracias a *jindexer* y ya está.



La ayuda se muestra de forma automática en navegador JavaHelp.

### Listado 3. Tabla de contenidos (primeraayudaTOC.xml)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE toc PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp TOC
Version 1.0//EN" "http://java.sun.com/products/javahelp/toc_1_0.dtd">
<toc version="1.0">
  <tocitem text="JavaHelp 2.0" target="principal" />
  <tocitem text="El fichero HelpSet" target="apartado1" />
  <tocitem text="El fichero de mapeado" target="apartado2" />
  <tocitem text="La tabla de contenidos" target="apartado3" />
  <tocitem text="La base de datos de búsquedas" target="apartado4" />
</toc>
```



Para ello solo debemos ir, en la línea de comandos, al directorio raíz donde están los ficheros HTML que componen la ayuda (generalmente ficheros HTML) y ejecutar el siguiente comando:

```
jhindexer *.html
```

Y se nos creará la base de datos ahí mismo, automáticamente. No hay que hacer nada más para tener las búsquedas habilitadas. Eso sí, si cambiamos los ficheros de ayuda deberemos ejecutar de nuevo este comando para actualizarla

## JAVAHHELPDEV

Con lo visto hasta ahora podemos realizar un sistema de ayuda online en Java de una forma sencilla, con los componentes básicos y más usuales. Aunque no es difícil realizar una ayuda con este procedimiento manual, es un poco pesado. Para facilitar esta labor, podemos descargar una herra-



JHelpDev nos ayuda a generar de forma sencilla los ficheros de la ayuda.

amienta gratuita y open source llamada jhelpdev (<http://jhelpdev.sourceforge.net/>), que permite crear todos los ficheros de forma sencilla, rápida y visual. Es una solución fácil de utilizar y no procede explicar aquí su uso, por lo que aconsejamos al lector probarla (incluida en el CD-ROM).

## INTEGRACIÓN EN UNA APLICACIÓN

Bien, ahora ya tenemos un sistema completo de ayuda definido, pero nos queda una cosa importante que hacer, integrar dicho sistema online dentro de nuestro proyecto Java. Es un proceso sencillo pero que requiere un poco de cuidado, así que lo vamos a ver con nuestro ejemplo. El código correspondiente se puede observar en el **Listado 5** (en el CD-ROM)

El listado muestra el código que se ejecuta al pulsar sobre el botón "Lanzar ayuda" de la ventana principal del ejemplo. Antes de nada, debemos tener la variable `CLASSPATH` de nuestro proyecto apuntando al fichero HelpSet de la ayuda. El segundo paso es importar dentro de nuestro proyecto java las clases de JavaHelp, lo cual hacemos mediante la inclusión de:

```
import javax.help.*;
```

dentro del código fuente. Y así podremos hacer uso de la API que proporciona JavaHelp 2.0. El código es realmente sencillo. Primero creamos un `ClassLoader` que nos permitirá posteriormente encontrar el fichero principal de la ayuda gracias a que éste estará en el `CLASSPATH` del proyecto

Luego, mediante el método `findHelpSet(...)` de la clase `HelpSet` y con ayuda del `ClassLoader` que hemos obtenido, conseguimos una URL normalizada que apunta al fichero HelpSet de nuestro sistema de ayuda. Además, este método se habrá encargado de comprobar mediante un `par` ser que el fichero está bien constituido y que, en principio, todo es correcto para poder lanzar el navegador de la ayuda.

En este momento se suelen producir muchos errores si es la primera vez que se usa JavaHelp, y casi todos son debidos a que no está correctamente definida la variable `CLASSPATH` y, por tanto, la instancia `ClassLoader` no es capaz de encontrar el fichero HelpSet que le indicamos. Paciencia y cuidado es lo único que hay que tener en este caso

El siguiente paso es crear una instancia de `JHelp` (el navegador de ayuda) al que le pasaremos por parámetros un objeto `HelpSet` creado con la URL obtenida y con la instancia de `ClassLoader` que hemos usado. Opcionalmente (no está de más), como es nuestro caso, podemos definir cuál de los identificadores que hemos especificado en el fichero de mapeado será el que se mostrará por defecto. Esto está bien para crear ayudas contextuales, es decir, que según donde estemos situados en nuestra aplicación, se pueda mostrar la ayuda concerniente al contexto actual, en lugar de una ayuda donde el usuario se tenga que poner a buscar

Y ya está. En este momento tenemos un navegador que muestra la ayuda creada sin ningún problema, pero todavía no se ve. Para ello habrá que insertarlo en alguno de los contenedores Swing de Java. En el ejemplo que estamos siguiendo, se crea un `JFrame` para ello, pero si nuestra aplicación tiene otro aspecto o un componente Swing ya preparado para insertar ahí la ayuda, se puede empotrar ahí sin problemas. `JHelp` implementa un `javax.swing.JComponent`, por lo que se puede insertar en cualquier lugar como si fuese un botón, un cuadro de texto o un panel

## CONCLUSIONES

Crear ayudas seguirá siendo, posiblemente, una de las tareas de menor agrado para el personal acostumbrado a trabajar con aspectos más técnicos del desarrollo software; pero por lo menos ahora podrá realizarse de una forma más cómoda, uniforme y eficiente en proyectos Java. Si el lector desea profundizar en este tema, quizás le resulte interesante el montar un servidor remoto de ayudas para JavaHelp o crear ayudas internacionalizadas, todo esto está bien documentado en la web de Javasoft.

### Listado 4. Fichero de índice (primeraayudaIndex.xml)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE index PUBLIC "-//Sun Microsystems Inc.//DTD JavaHelp Index
Version 1.0//EN" "http://java.sun.com/products/javahelp/index_1_0.dtd">
<index version="1.0">
  <indexitem text="Análisis del lenguaje natural" target="apartado4"/>
  <indexitem text="Ayuda online" target="principal"/>
  <indexitem text="Base de datos" target="apartado4"/>
  <indexitem text="Búsqueda avanzada" target="apartado4"/>
  <indexitem text="Definición de vistas" target="apartado1"/>
  <indexitem text="Enlaces" target="apartado2"/>
  <indexitem text="Fichero de contenido" target="apartado1"/>
  <indexitem text="Fichero de mapeado" target="apartado1"/>
  <indexitem text="HelpSet" target="apartado1"/>
  <indexitem text="ID's" target="apartado2"/>
  <indexitem text="JavaHelp 2.0" target="principal"/>
  <indexitem text="Jindexer" target="apartado4"/>
  <indexitem text="Navegador JavaHelp" target="principal"/>
  <indexitem text="Sistema integrado" target="principal"/>
  <indexitem text="Tabla de contenidos" target="apartado3"/>
  <indexitem text="Árbol de contenidos" target="apartado3"/>
</index>
```



1ª entrega, DVD-ROM  
y Presentación de la obra  
por solo **5,99** euros



# USUARIO LINUX

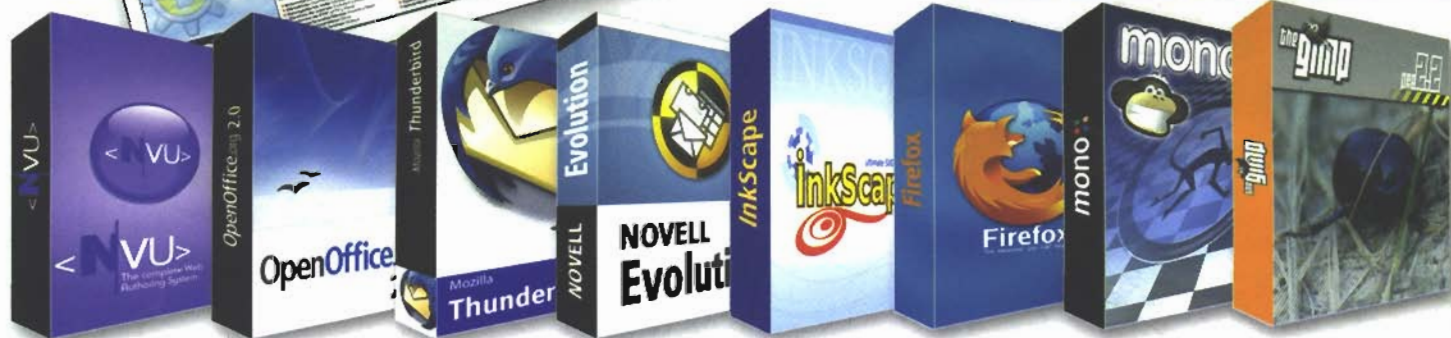


► **Usuario Linux** es una obra compuesta por doce entregas de periodicidad semanal, cuyo objetivo es formar al lector en el manejo del sistema Linux y sus principales aplicaciones

► Aprende a trabajar de manera práctica, con tutoriales guiados y explicaciones sencillas, con el sistema operativo de mayor futuro, la plataforma que ya utilizan las empresas

► Linux es el sistema más moderno y eficiente para PC, con él olvídate de cuelgues, virus, spyware y demás problemas

► Con el número 1 se incluye un DVD con Debian Linux Sarge, un sistema Linux completo y en español que incluye más de 7000 aplicaciones.



<b>NVU</b> Diseño Web	<b>OpenOffice.org</b> Suite de Ofimática	<b>Thunderbird</b> Gestión de Correo	<b>Evolution</b> Suite de Comunicaciones	<b>Inkscape</b> Autoedición de Textos	<b>Firefox</b> Navegador Web	<b>Mono</b> Desarrollo .NET	<b>Gimp</b> Tratamiento y Retoque Digital
--------------------------	---	---	---	--	---------------------------------	--------------------------------	--

## Composición de la obra

12 coleccionables divididos en:

### Teoría

Conoce el sistema Linux a nivel profesional, cómo se instala en un PC, cómo se administra y cómo se trabaja con los principales entornos gráficos.

### Software

Descubre todas las aplicaciones libres que hay para Linux, desde la suite de ofimática OpenOffice.org a las herramientas de desarrollo compatibles con .NET.

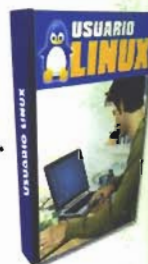
### Tutorial Práctico

Aprende lo que realmente se necesita hoy en día: Cómo montar un servidor web, cómo configurar una red local, cómo programar una web dinámica o cómo instalar un cortafuegos profesional.

## Oferta de suscripción

**Usuario Linux** está disponible en tu quiosco, pero también puedes suscribirte directamente llamando al **91 628 02 03** y recibir la obra en tres entregas mensuales por solo **66,00** euros.

Además conseguirás de regalo la tapa para encuadernar la obra y una minisuscripción de tres números a la revista **Todo Linux**.



**Studio  
PRESS**

C/ del Río Ter, Nave 13 • Polígono Industrial "El Nogal" • 28110 Algete (Madrid) • Tel.: 916280203 • Fax: 916280935  
e-mail: [usuario@iberprensa.com](mailto:usuario@iberprensa.com) • [www.studiopress.es](http://www.studiopress.es)

**ya a la venta en quioscos y centros comerciales**